Neural Machine Translation with RNNs

(1-g)

i. 掩码作为标记是否为填充标记的指示符。在计算注意力时，我们可以使用 masked fill() 有效地将 pad mask == 1 的标记的注意力值设置为-inf。在对该向量应用 softmax 时，注意力值为 inf 的标记的 softmax 值将为 0，从而被注意力机制 “忽略”。

ii. 在注意力机制中忽略填充标记是非常重要和直观的，因为这些标记没有实际价值，甚至会对注意力计算造成不必要的干扰。更糟糕的是，这些标记可能会在解码阶段被预测出来。当我们需要对序列中的特定标记进行填充时，填充掩码的实现既灵活又高效。

(1-i)

点积 vs 乘积

- 优点：需要训练的参数少，时间和内存成本低。
- 缺点：由于缺乏可训练的权重矩阵，灵活性较差。

加法 vs 乘法

- 优点：加法结果的规模更可控，因为应用乘法比应用加法更容易获得巨大的数值。
- 缺点：与乘法运算相比，加法运算需要更多时间，因为 Python 对矩阵乘法进行了专门优化。

Analyzing NMT Systems

(2-a)

由于用户可以根据不同的词素组合创造出新的复杂单词，这就很容易在使用词级嵌入时产生词汇遗漏（Out-Of-Vocabulary，OOV）问题。利用子词级学习不仅可以缓解 OOV 问题，还能确保拼写相似的单词在向量空间中的嵌入度相互接近。尤其是对于切罗基语来说，单词每个字符都表示一部分含义，所以使用字符级的编码更合适。

(2-b)

如问题(2-a)，单词的前缀也表达了一定的意思，所以使用字符级的编码。

(2-c)

使用迁移学习应该可以帮助缓解。

(d)

i.代词错误，类似于英语中的一词多义（aunt=阿姨，伯母…）

ii.遇到了一个训练集中不存在的切罗基语，可能可以使用迁移学习或其他方式扩充数据集。

iii.固定搭配翻译错误，采用强制引入规则的方法克服。

（f）

i. c1: p1 = 0.9231, p2 = 0.8333 len(c) = 13, len(r) = min(13, 14) = 13, BP = 1 BLEU score = 0.8771 - c2: p1 = 0.8462, p2 = 0.75 3 len(c) = 13, len(r) = min(13, 14) = 13, BP = 1 BLEU score = 0. 7966 根据 BLEU 得分，第一个预测被认为是更好的预测，因为它准确地涵盖了更多的 n-gram 词组，如 "in the "和 "not comprehend"。

ii. 对于 c1：BLEU 得分 =0.7161；对于 c2：BLEU 得分 =0.7966 如果我们只关注 BLEU 得分，那么第二个预测被认为是更好的预测，因为它命中了更多的 n-gram 标记。但是，如果我们用人眼看一下预测结果，就会发现与第一个预测结果相比，第二个预测结果存在语法不连贯的问题。更重要的是，第二次预测还产生了 "trails "等不相关的词。

iii. 在机器翻译任务中，我们应该更关注预测是否保留了源句的完整语义信息，而不是判断它是否包含特定的单词/标记。在很多情况下，我们可以使用多个意译或同义词来表达相同的语义，这正是使用多个参考文献的意义所在。一般来说，使用多个参考文献比只使用一个参考文献能获得更准确、更稳健的评估结果。

iv.

优点：

易于计算，效率高，可节省人力成本。
与语言无关，减少了对领域专家的需求。
缺点：
不考虑语法正确性。
不考虑同义词或类似表达